SH7145F

Synchronous Serial Data Transmission

Summary

The SH7144 series is a single-chip microprocessor based on the SH-2 RISC (Reduced Instruction Set Computer) CPU core and integrating a number of peripheral functions.

This application note describes synchronous serial data transmission using the SCI (Serial Communication Interface) module of the SH7145F. It is intended to be used as reference by users designing software applications.

The program examples contained in this application note have been tested. However, operation should be confirmed before using them in an actual application.

Device for Which Operation Has Been Confirmed

SH7145F

Contents

1. Specifications 2
2. Functions Used 3
3. Operation 6
4. Software 7
5. Flowcharts 10
6. Program Listing 13

1. Specifications

As shown in figure 1, synchronous serial data transmission is performed using channel 0 (ch0) of the SCI module of the SH7145F. In this task example 3 bytes of serial data are transmitted from the SH7145. The communication format is 8-bit fixed data length, LSB first, with a logical bit rate of 250,000 bps.

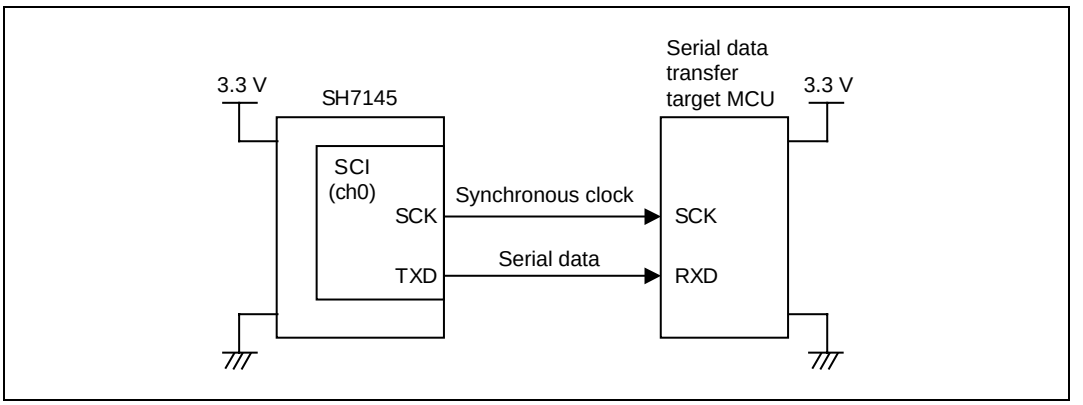


Figure 1 Synchronous Serial Data Transmission by SH7145

2. Functions Used

In this task example the SCI (Serial Communication Interface) is used to perform synchronous serial data transmission. Figure 2 shows a block diagram of channel 0 (ch0) of the SCI module. The functions of the elements shown in figure 2 are described below.

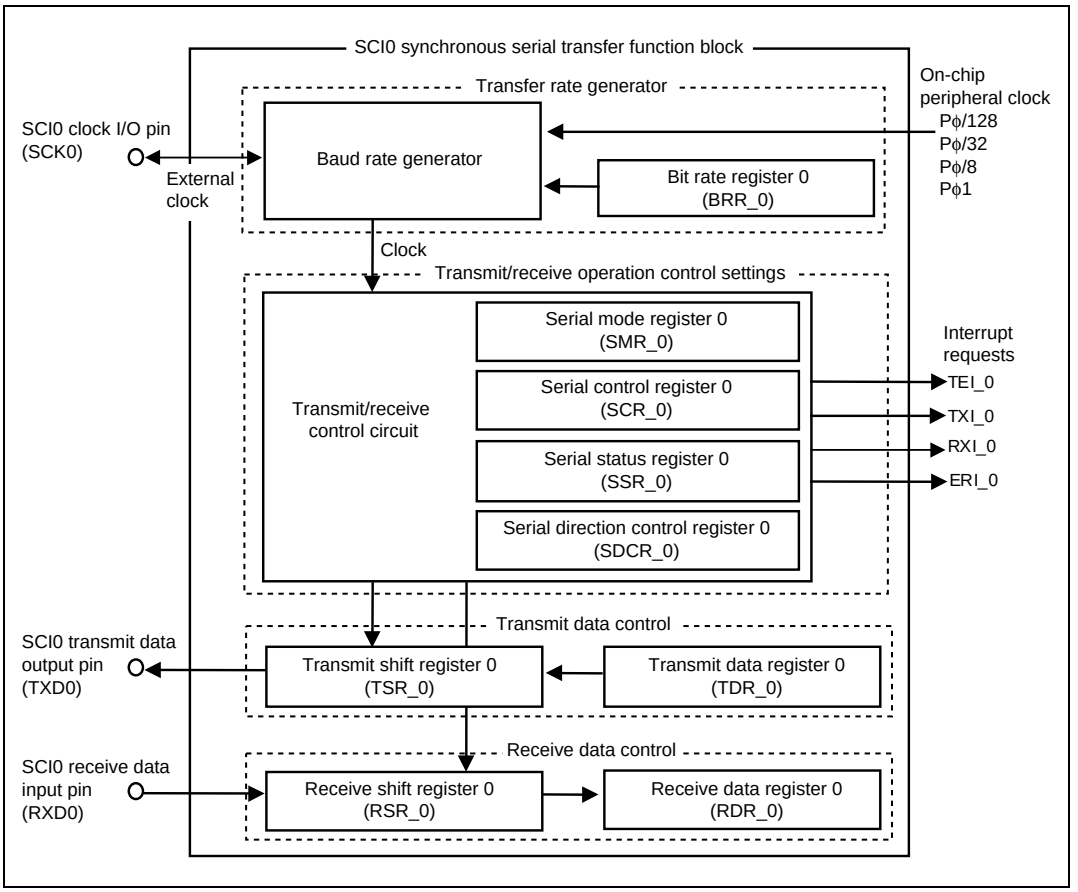


Figure 2 SCI (ch0) Block Diagram

- Synchronous Mode

  Data is transmitted and received in synchronization with clock pulses. The transfer data length is fixed at 8 bits.

- On-Chip Peripheral Clock Pϕ

  This is the reference clock for operation of on-chip peripheral functions. The clock signal is generated by a clock oscillator.

- Receive Shift Register (RSR_0)

  This register is used to receive serial data. Serial data is input to RSR_0 from the RxD_0 pin. When one frame of data has been received, it is automatically transferred to the receive data register (RDR_0). RSR_0 cannot be accessed by the CPU.

- Receive Data Register (RDR_0)

  Received data is stored in this 8-bit register. When one frame of data has been received, it is automatically transferred from RSR_0. RSR_0 and RDR_0 are in a double-buffer configuration, allowing continuous reception of data. RDR_0 is a receive-only register, so it can only be read by the CPU.

- Transmit Shift Register (TSR_0)

  This register is used to transmit serial data. In order to transmit data, the data is first transferred from the transmit data register (TDR_0) to TSR_0. Then the transmit data is output from the TxD_0 pin. TSR_0 cannot be accessed directly by the CPU.

- Transmit Data Register (TDR_0)

  Data to be transmitted is stored in this 8-bit register. When it is detected that TDR_0 is empty, data that has been written to TDR_0 is automatically transferred to TSR_0. TDR_0 and TSR_0 are in a double-buffer configuration. This allows data to be transferred to TSR_0 after one frame of data has been transmitted and the next frame of data is still being written to TDR_0, making possible continuous transmission of data. It is always possible to read or write to the TDR from the CPU, but before writing to the TDR it should be confirmed that the value of the TDRE bit in the serial status register (SSR_0) is 1.

- Serial Mode Register (SMR_0)

  This 8-bit register is used to select the serial data communication format and the clock source for the on-chip baud rate generator.

- Serial Control Register (SCR_0)

  This register is used for transmit and receive control, interrupt control, and to select the transmit and receive clock source.

- Serial Status Register (SSR_0)

  This register comprises the SCI0 status flag and the transmit and receive multiprocessor bits. TDRE, RDRF, ORER, PER, and FER can be cleared only.

- Serial Direction Control Register (SDCR_0)

  This register is used to select whether the LSB or MSB is first. For 8-bit communication either LSB-first or MSB-first may be selected, but LSB-first should be used for 7-bit communication.

- Bit Rate Register (BRR_0)

  This 8-bit register is used to adjust the bit rate. The SCI has independent baud rate generators for the individual channels, allowing different bit rates to be set for each. See the hardware manual for details on setting values, execution rate relationships, etc.

Table 1 shows the function allocations for the task example.

Table 1 Function Allocations

Function	Classification	Function Allocation
SCK0	Pin	Channel 0 synchronous clock I/O
TXD0	Pin	Channel 0 transmit data output pin
RXD0	Pin	Channel 0 transmit data input pin
SMR_0	SCI0	Sets communication format to synchronous mode
SCR_0	SCI0	Enables transmit operation
SSR_0	SCI0	Status flag showing SCI0 operation status
SDCR_0	SCI0	Specifies LSB-first
BRR_0	SCI0	Sets communication bit rate
TSR_0	SCI0	Register for transmitting serial data
TDR_0	SCI0	Register for storing transmit data

3. Operation

Figure 3 shows the operation of synchronous mode data transmission in the task example. To help explain figure 3, table 2 lists the software and hardware processing that is performed.

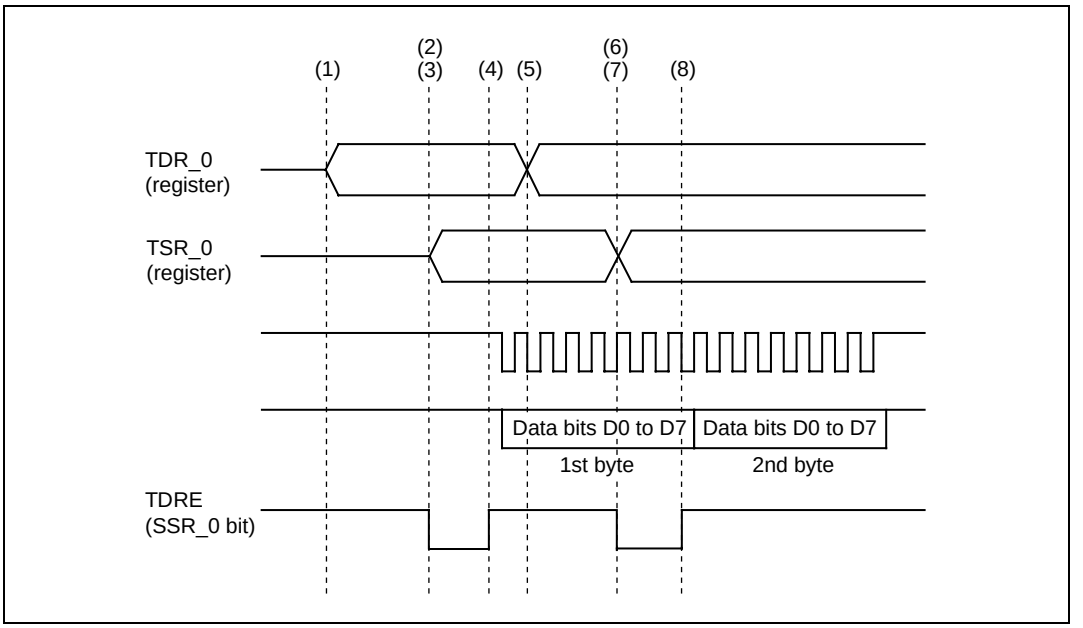


Figure 3 Data Transmission Operation

Table 2 Processing

	Software Processing	Hardware Processing
(1)	Write transmit data to TDR_0	—
(2)	Clear TDRE flag in SSR_0 to 0	—
(3)	—	Transfer data from TDR_0 to TSR_0
(4)	—	Set TDRE flag in SSR_0 to 0 and output transmit data from pin TXD0
(5)	Confirm TDRE is 0 and send next transmit data to TDR_0	—
(6)	Clear TDRE flag in SSR_0 to 0	
(7)	—	Transfer data from TDR_0 to TSR_0
(8)	—	When output of previous data finishes, set TDRE flag in SSR_0 to 1 and output transfer data from pin TXD0
(9)	Repeat	Repeat

4. Software

(1) Module Descriptions

Table 3 lists the modules used in the task example.

Table 3 Module Descriptions

Module	Label	Function
Main routine	main	Calls modules
SCI routine	init_sci	Initial settings of SCI0
Transfer routine	trans_sci	Transmits serial data

(2) Argument Descriptions

Table 4 lists the arguments used in the task example.

Table 4 Argument Descriptions

Argument	Function	Module
TRANS_DATA_1	Transmits data from SCI_0	Transfer routine
TRANS_DATA_2	Transmits data from SCI_0	Transfer routine
TRANS_DATA_3	Transmits data from SCI_0	Transfer routine

(3) On-Chip Register Descriptions

Table 5 lists the on-chip registers used in the task example. The set values shown are the values used in the task example and differ from the initial settings.

Table 5 On-Chip Register Descriptions

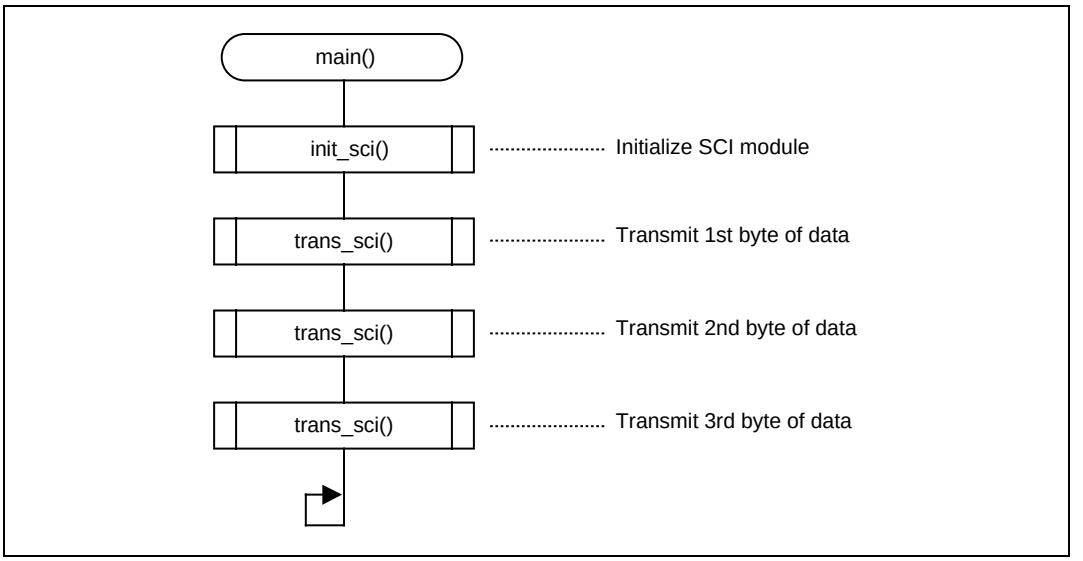
Register	Bit	Set Value	Function
MSTCR1	MSTP16	0	Module standby control register 1 SCI0 standby control bit Standby cancelled when MSTP16 = 0
SCR_0		H'20	Serial control register 0 (SCI_0) Transmit and receive control, interrupt control, transmit and receive clock source control
	TIE	0	Transmit interrupt enable TXI interrupt requests enabled when set to 1
	RIE	0	Receive interrupt enable RXI and ERI interrupt requests enabled when set to 1
	TE	1	Transmit enable Transmit operations enabled when set to 1
	RE	0	Receive enable Receive operations enabled when set to 1
	MPIE	0	Multiprocessor interrupt enable (In asynchronous mode, enabled when MP = 1 in SMR) In the task example, disabled because MP = 0
	TEIE	0	Transmit end interrupt enable TEI interrupt requests enabled when set to 1
	CKE1 CKE2	0 0	Clock enable 1, 0 Selects clock source and SCK pin function In the task example, clock source is on-chip clock and SCK pin is not used
SMR_0		H'00	Serial mode register 0 Selects communication format and the clock source for on-chip baud rate generator
	C/A	0	Communication mode Synchronous mode when cleared to 0
	CHR	0	Character length (enabled in asynchronous mode only) Disabled in task example because synchronous mode is used
	PE	0	Parity enable (enabled in asynchronous mode only) Disabled in task example because synchronous mode is used
	O/E	0	Parity mode (enabled in asynchronous mode when PE = 1) Disabled in task example because synchronous mode is used
	STOP	0	Stop bit length (enabled in asynchronous mode only) Disabled in task example because synchronous mode is used

Register	Bit	Set Value	Function
SMR_0	MP	0	Multiprocessor mode (enabled in asynchronous mode only) Disabled in task example because synchronous mode is used
	CKS1 CKS2	0 0	Clock select 1, 0 When value is 00, Pϕ clock selected using on-chip baud rate generator as clock source
BRR_0		H'40	Bit rate register 0 8-bit register for adjusting bit rate
SDCR_0		H'F2	Serial direction control register 0 DIR bit (bit 3) selects LSB-first or MSB-first In task example, DIR = 0 (LSB-first)
SSR_0		H'xx	Serial status register 0 Comprises SCI0 status flag and transmit and receive multiprocessor bits Only 0 may be written to the status flag, to clear it
	TDRE	*	Transmit data register empty (status flag)
	RDRF	*	Receive data register full (status flag)
	ORER	*	Overrun error (status flag)
	FER	*	Framing error (status flag)
	PER	*	Parity error (status flag)
	TEND	*	Transmit end (status flag)
	MPB	0	Multiprocessor bit
	MPBT	0	Multiprocessor bit transfer
PACRL2	PA2MD1 PA2MD0	0 1	Port A control register L2 Function setting for port A multiplex pin (SCK0)
	PA1MD1 PA1MD0	0 1	Port A control register L2 Function setting for port A multiplex pin (TXD0)

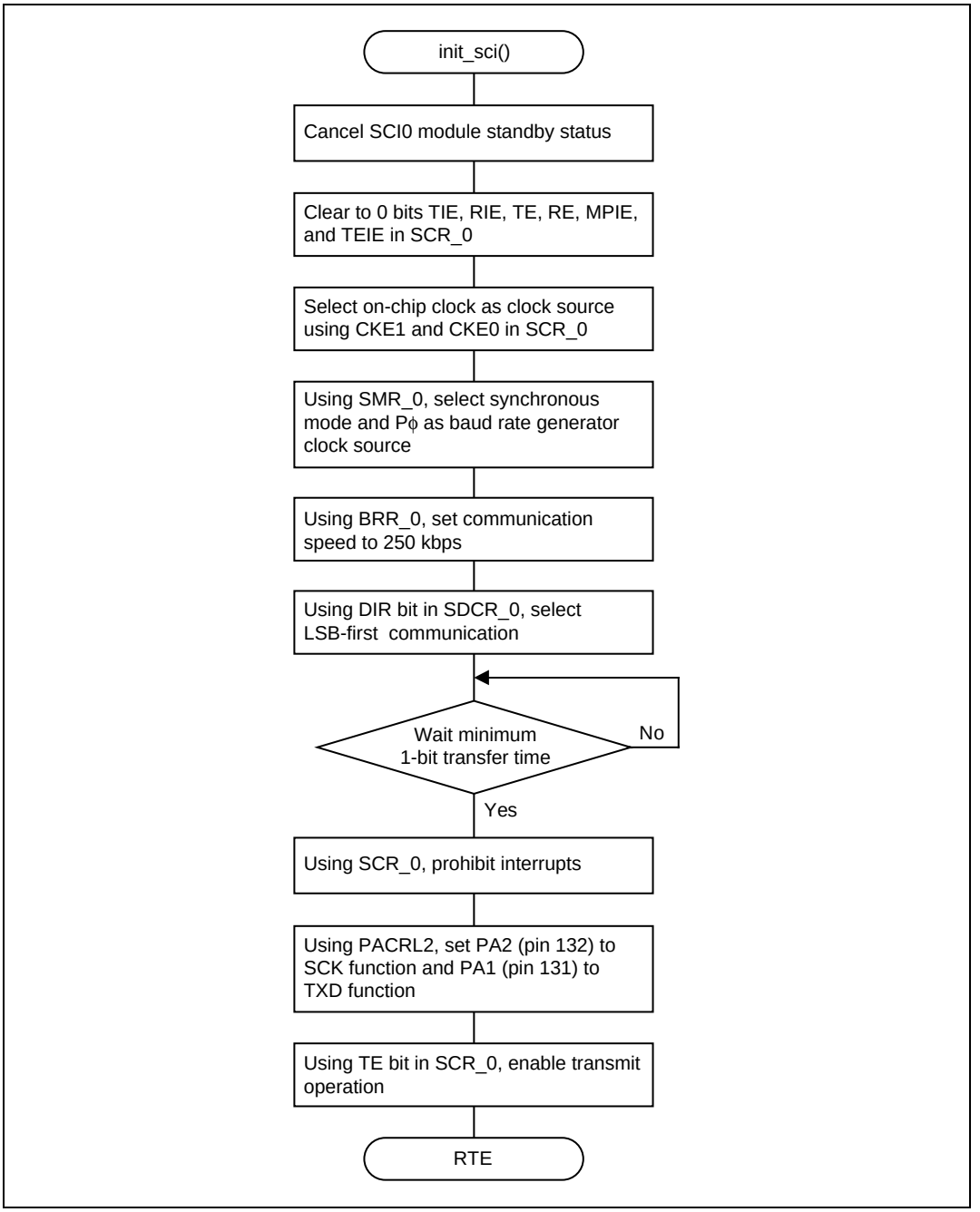
*: Can only be cleared to 0. Setting to 1 is performed by hardware.

5. Flowcharts

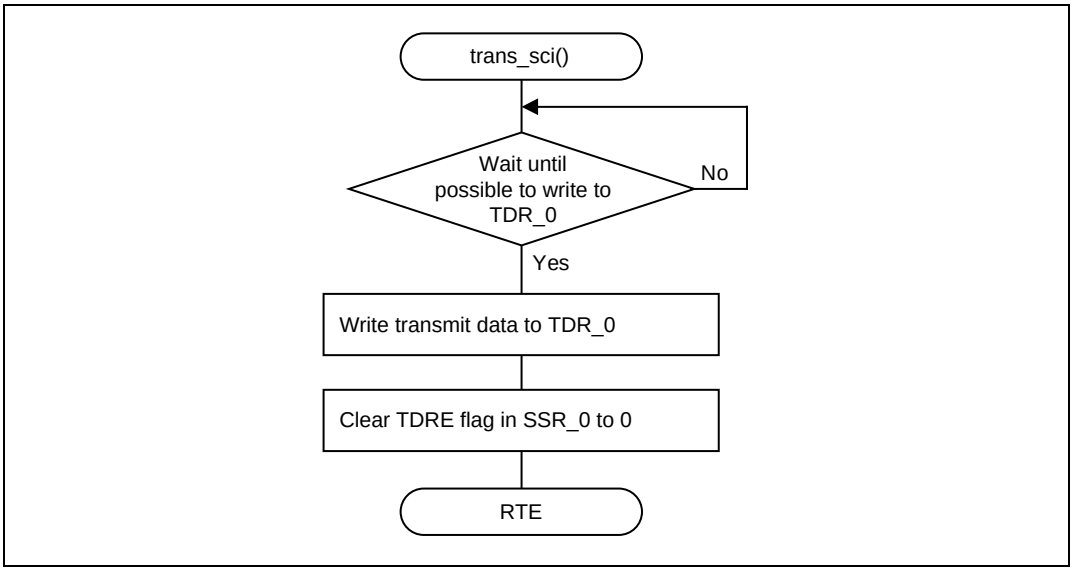
(1) Main Routine



(2) SCI1 Initialize Routine


init_sci()
Cancel SCI0 module standby status
Clear to 0 bits TIE, RIE, TE, RE, MPIE, and TEIE in SCR_0
Select on-chip clock as clock source using CKE1 and CKE0 in SCR_0
Using SMR_0, select synchronous mode and Pϕ as baud rate generator clock source
Using BRR_0, set communication speed to 250 kbps
Using DIR bit in SDCR_0, select LSB-first communication
Wait minimum 1-bit transfer time
No
Yes
Using SCR_0, prohibit interrupts
Using PACRL2, set PA2 (pin 132) to SCK function and PA1 (pin 131) to TXD function
Using TE bit in SCR_0, enable transmit operation
RTE

(3) Data Transmit Routine



6. Program Listing

```
/************************************************************/
/* SH7145F Application Note                                 */
/*                                                          */
/* Function                                                 */
/*       :SCI0                                              */
/*       :Clock Synchronous Transmit Mode                   */
/*                                                          */
/* External input clock       :12.5MHz                      */
/* Internal CPU clock         :50MHz                        */
/* Internal peripheral clock  :25MHz                        */
/*                                                          */
/* Written                    :2003/7     Rev.1.0           */
/************************************************************/

#include "iodefine.h"
#include <machine.h>

/*------------ Symbol Definition ---------------------------------------------*/

#define TRANS_DATA_1   0x61
#define TRANS_DATA_2   0x62
#define TRANS_DATA_3   0x63

/*------------ Function Definition -------------------------------------------*/
void main(void);

void init_sci(void);
void trans_sci(char c);

void dummy_f(void);

/*------------ RAM allocation Definition -------------------------------------*/

/************************************************************/
/* main Program                                             */
/************************************************************/
void main( void )
{
   unsigned char i = 0;

   init_sci();                 /* Initialize SCI              */

   trans_sci(TRANS_DATA_1);    /* Transmit 1st byte of data   */

   trans_sci(TRANS_DATA_2);    /* Transmit 2nd byte of data   */

   trans_sci(TRANS_DATA_3);    /* Transmit 3rd byte of data   */
```

```
   while(1);                    /* LOOP                          */
}

/************************************************************
  Function  : init_sci
  Operation : Initialize serial (sci0)
       sci0 : Synchronous transmit operation
************************************************************/
void init_sci(void)
{
   unsigned long i;

   P_STBY.MSTCR1.BIT.MSTP16 = 0;   /* disable SCI0 standby mode       */

   /* Initialize SCI      */
   P_SCI0.SCR_0.BYTE &= 0x03;      /* clear TIE,RIE,TE,RE,MPIE,TEIE   */
   P_SCI0.SCR_0.BIT.CKE = 0;       /* clock:internal,SCK:output       */
   P_SCI0.SMR_0.BYTE = 0x80;       /* Clock synchronous mode          */
                 //   CA = 1;     /* Clock synchronous mode          */

   P_SCI0.BRR_0 = 24;              /* 250kbps@25MHz(Peripheral)       */
   P_SCI0.SDCR_0.BIT.DIR = 0;      /* LSB first send                  */

   for( i=0; i < 0x0300 ; i++);    /* Wait 1bit (over 4usec)          */

   /* Initialize SCI0 PORT         */
   P_PORTA.PACRL2.BIT.PA2MD = 1;   /* set SCK0(PA2:132pin@SH7145)     */
   P_PORTA.PACRL2.BIT.PA1MD = 1;   /* set TXD0(PA1:131pin@SH7145)     */

   P_PORTA.PAIORL.BIT.PA2IOR = 1;  /* SCK0 Output                     */

   P_SCI0.SCR_0.BIT.TE = 1;        /* TE=1,Transmit Enable(SCI0)      */

}

/*********************************************************************/
/* Function       : trans_sci                                        */
/* Operation      : Write 1 character to serial output               */
/* Argument       : c                                                */
/* Value returned : None                                             */
/*********************************************************************/
void trans_sci(char c )
{
   while(!(P_SCI0.SSR_0.BYTE & 0x80)); /* Wait until data can be written to TDR */

   P_SCI0.TDR_0 = (unsigned char)c;    /* Write data to TDR           */
   P_SCI0.SSR_0.BYTE &= 0x7F;          /* Clear flag, transmit        */
}
```

```
/************************************
          Interrupt handling
************************************/
#pragma interrupt(dummy_f)
void dummy_f(void)
{
        /*   Other Interrupt   */
}
```

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.